Dragons à foison

Jean-Eric SCHOETTL

Il y a un mois mon patron Lee Norman me convoque dans son somptueux bureau.

" Une affaire pour vous, Jess. Vous vous souvenez de notre vieil ami le Dragon ?"

La tronche du vieil ami apparaît sur l'écran:

$$\begin{cases} m_0 = \Lambda \\ m_{i+1} = m_i \; a(i) \; \widetilde{\overline{m}}_i \end{cases}$$

Un dragon $\longleftrightarrow$ une fonction $a \in 2^{\mathbb{N}}$

" Regardez le bien, Jess. N'y manque-t-il pas quelque chose ?
- Je vous suis 5 sur 5 boss. On pourrait en faire un genre de mot de Dyck en complétant la partie droite de la règle de production par la bonne parenthèse.
- Juste."

Et il projette la définition rafistolée:

$$\begin{cases} m_0 = \Lambda \\ m_{i+1} = m_i \; a(i) \; \widetilde{\overline{m}}_i \; \overline{a(i)} \end{cases} \qquad (a \in 2^{\mathbb{N}})$$

" Chaque mot restant préfixe des suivants, l'interprétation graphique en gauche-droite reste possible. Vous voyez où je veux en venir ?
- J'y suis! La propriété de non-intersection! Est-elle conservée ?
- C'est exactement ce que je veux savoir."

Bon.Une nouvelle mission. Je mets mon pote Harald dans le coup.

Jamais trop prudent avec les dragons.

Quinze jours plus tard, les doigts encore souillés d'encre à traçante, nous nous repointons au département.

" Alors ?
- Alors ...... évidemment, c'est des dragons.

Qui génèrent sec même:
Ca nous a fait arrêter à la douzième génération, vu qu'on stocke un bit par octet T 1600.

- Pas moyen d'éviter de stocker ?
- On en a pas trouvé en tout cas."

Ca s'annonce mal. Mais le plus dur n'est pas encore passé.

" Comment avez-vous procédé ?
- On a fait ça en ASM. On sort le binaire et les directions sur imprimante, l'interprétation graphique sur table traçante. Ah! On arrondit les angles ..."

Ca n'arrondit pas les siens en tout cas.

" Qu'avez-vous testé ?
- Le programme est écrit une fois pour toutes. Pour chaque membre de la famille l'utilisateur n'a qu'à rédiger le sous-programme qui génère a (i) dans l'accumulateur à chaque génération.
- Résultats ?
- On aurait eu l'air plus fins s'ils avaient été complétement négatifs. Mais là c'est râlant: ça se coupe pas pour le "flip-flop", ça se coupe pas pour le "dragon qui se mange la queue". Par contre, s'il est autophage de 2 en 2, par exemple, on observe des superpositions - oh pas ben grosses - mais qui suffisent à démolir votre conjecture, patron."

On lui montre le dossier: DRAGON DYCK, le listing du programme ainsi que quatre "nouveaux dragons" avec les débuts de leurs sorties binaires et les sous-programmes FFL qui leur sont associés:

(1) a (i) = 1 $\forall i$

Vous croyez reconnaître? Vous vous trompez: il manque une patte de mouche pour que ce soit un vieux dragon.

(2) Flip-Flop

a (i) = Si (i pair) alors 0 sinon 1

C'est notre petit Noël à nous.

(3) Se bouffe la queue

a (i) = Si (i = 1) alors 0 sinon $i^{ème}$ bit de m

(4) Se bouffe la queue de 2 en 2

a (i) = Si (i = 1) alors 0 sinon 2 (i - 1ème) bit de m

Sur les 4, seul le dernier se coupe.

D'autres dragons ont été tracés: autophages en progression arithmétique, géométrique, variable. En général, il y avait des chevauchements.

Pas besoin de préciser que tout ceci est ultra-confidentiel ...

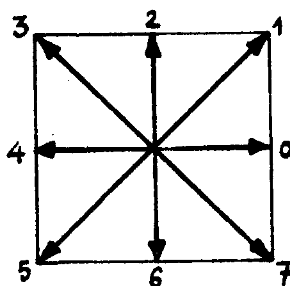
INFORMATIONS PRATIQUES

Ordinateur: Télémécanique T 1600 sous BOS D, machine à mots, octets adressables en mode indirect post indéxé.

Table traçante: calcomp

Langage utilisé: assembleur ASM.

Programme de traçage PASELN (de Claude LEBRUN) commande la table traçante. Il comporte à chaque appel 2 paramètres: le premier est le nombre de pas (1 pas droit élémentaire = 1/10 mm) le second la direction selon le schéma ci-dessous. Ces deux paramètres sont rangés respectivement dans les registres accumulateur A et index X.



Les valeurs 8 et 9 du paramètre "direction" sont réservées respectivement au lever et baisser de plume.

Un pas oblique = un pas droit * $\sqrt{2}$

```
Exemple: APASLN:   WORD   PASELN
                   LAI    0
                   LXI    6
                   BSR    APASLN
```

Ceci va entraîner un déplacement de la plume de 0, 6 mm vers la droite.

Programme DRAGON DYCK

Il comprend deux parties, un noyau commun à tous les nouveaux dragons et un sous programme FFL propre à chacun de ces dragons qui calcule a (i) à chaque génération.

Le programme DRAGON DYCK itère sur la séquence suivant :

- élaboration du prochain bit a (i); stockage d'icelui dans BITA et de son complémentaire dans BITB (étiquette RE)

- Génération de l'image-miroir complémentée $\overset{\backsim}{\overline{m}}_i$ (étiquette ETI)
- Rangement de BITB à la suite (étiquette ETI1)
- Interprétation du suffixe a (i) $\overset{\backsim}{\overline{m}}_i$ $\overline{a(i)}$ (impression des directions et du binaire, traçage de la courbe) (étiquette ETI2)





```
        IDP       "DRAGON DICK"
        < VALS I/O
        EXT PASELN                     < SOUS PROGRAMME DE TRACAGE
OUT:    VAL       '2000                <  SORTIE
LO:     VAL       '86                  < IMPRIMANTE
EMOD:   VAL       '8000                < RETOUR EN FIN D ECHANGE
IOCS:   VAL       '8                   < SYSTEME DE GESTION DES I/O
ABOS:   VAL       'C                   < SUPERVISEUR
        <
        <
        COMMON
        <
DEBCOM:EQU        $+128
        <
        < REGISTRES DE TRAVAIL
        < INDEX SUR TAB
IND:    WORD      -1
RAJ:    WORD      0
INDX:   WORD      -1
COMPT:  WORD      0
        <  BITS ARTICULATOIRES
BITA:   WORD      0                    <   BIT A(I)
BITB:   WORD      1                    < BIT A(I) NIE
            < DIRECTIONS
OLD:    WORD      2
OLD1:   WORD      0
DIR:    BYTE      6;2;0;4;2;6;4;0 < DIRECTIONS DROITES

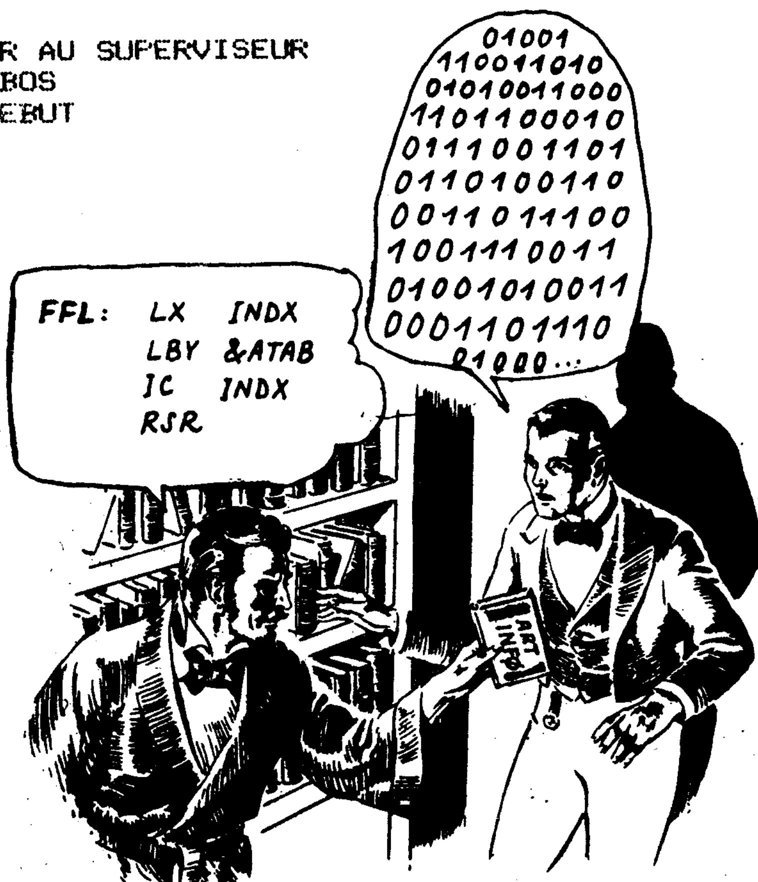

DIR1:   BYTE      7;1;1;3;3;5;5;7 < DIRECTIONS OBLIQUES


            < CONSTANTES
MAX:    WORD      4000                 < TAILLE MAXIMUM DU DRAGON
N34:    WORD      34
APILE:  WORD      PILE-1               < INITIALISATION DU POINTEUR PILE
        <
            < SOUS - PROGRAMMES
AFFL:   WORD      FFL                  < SP SPECIFIQUE A CHAQUE DRAGON:GENERE A(I)
ATRACE:WORD       TRACE                < CALCUL DE LA NOUVELLE DIRECTIN
APASLN:WORD       PASELN               < PLOTTER
            <<
            << POINTEURS DE TABLEAUX
ATAB:   WORD      TAB,X
ADIR:   WORD      DIR,X
ADIR1:  WORD      DIR1,X
ABUF:   WORD      BUF,X
            <<
            <<TABLE DE PARAMETRES POUR LA SORTIE DU BINAIRE
IOCB:   WORD      OUT+EMOD+LO
        WORD      BUF
        WORD      120
        DZS       2
```

```
        <<
        <<
      TABLE
      WORD      0
TAB:  DZS       4000          <     STOCKAGE DU CORPS DU DRAGON
FILE: DZS       20            <   ZONE PILE
BUF:  DZS       59            <BUFFER DE SORTIE
      WORD      '8D0A         <    PASSAGE A LA LIGNE
        <<
        <<
      PROG
      <
        < INITIALISATION DES REGISTRES DE BASE ET DE PILE
      WORD      DEBCOM
DEBUT: LRP      C
      LY        -1,C
      LR        Y,C
      LA        APILE
      LR        A,K
        <  EST-CE FINI?
RE:   LA        IND
      CP        MAX
      JG        SORTIE
      <
        < GENERATION DU PROCHAIN BIT A(I) ET DE SON COMPLEMENTAIRE
      BSR       AFFL
      STA       BITA
      IBT       15
      STA       BITB
      LA        BITA
      LB        IND
      ADRI      1,B
      LR        B,X
      STBY      &ATAB
        <
        < INITIALISATION DES INDEX RAJ ET IND POUR LA GENERATION DE
        < M(I)  BARRE TILDE
      LAI       1
      STA       RAJ
      STX       IND
      <
        <  M (I) BARRE TILDE
      <
        < TROUVER LE COMPLEMENTAIRE DE TAB(IND - RAJ)
ETI:  LA        IND
      LY        RAJ
      SBR       Y,A
      LR        A,X
      JAL       ETI1
      LBY       &ATAB
      IBT       15
        <
        < LE METTRE DANS TAB(IND + RAJ)
      LX        IND
      ADR       Y,X
```

```
         STBY     &ATAB
            <
            <    BOUCLE
         IC       RAJ
         JMP      ETI
            <
            <  METTRE BITB DANS TAB JUSTE APRES M(I) A(1) M(I) BARRE T1LDE
ETI1:    LX       IND
         ADR      Y,X
         LA       BITB
         STBY     &ATAB
            <
            <INTERPRETATION GRAPHIQUE ET SORTIE BINAIRE DE CE QUI VIENT
            <  D ETRE GENERE
            <
            < RAJ DEBUTE ET IND TERMINE LA SEQUENCE GENEREE
         LA       IND
         STA      RAJ
         STX      IND
         <
            < A-T-ON FINI D'INTERPRETER?
ETI2:    LA       RAJ
         CP       IND
         JG       RE
            < TRACER L'ANGLE ET LE PROCHAIN SEGMENT
         BSR      ATRACE
         LA       OLD1
         LXI      6
         BSR      APASLN
         LA       OLD
         LXI      8
         BSR      APASLN
            <
            <  ENTREPOSER LA NOUVELLE DIRECTION
         LA       OLD
         PTY
         LAI      0
         ADCR     A
         JAE      $+2
         LAI      '80
         AD       OLD
         ADRI     '30,A
         LX       COMPT
         STA      &ABUF
               <EST IL TEMPS D IMPRIMER LA LIGNE?
         IC       COMPT
         LAI      25
         CP       COMPT
         JG       ETI2
         STZ      COMPT
         LAD      IOCB
         SVC      IOCS
            < BOUCLE
         JMP      ETI2
         <
```

```
         <
         <
         < ENTREPOSER LE BIT COURANT DANS LE BUFFER DE SORTIE ET CALCUL
         <LE TOURNANT ET LA NOUVELLE DIRECTION
TRACE:  LX      RAJ             <METTRE LE CODE INTERNE DE 0 OU 1 DANS Y
        LBY     &ATAB
        LR      A,Y
        ADRI    '30,Y
        JAE     $+2
        LYI     'B1
        LX      COMPT           < ENTREPOSER
        LB      N34
        ADR     B,X
        STY     &ABUF
        LX      OLD              <PROCHAINE DIRECTION DANS OLD
        ADR     A,X
        LBY     &ADIR
        STA     OLD
        LBY     &ADIR1          <PROCHAIN ANGLE DANS OLD1
        STA     OLD1
        IC      RAJ
        RSR
          <
        <
          <  PREMIER DRAGON
FFL:    LAI      1
        RSR
        <
          <
          <RETOUR AU SUPERVISEUR
SORTIE:SVC      ABOS
        END     DEBUT
```

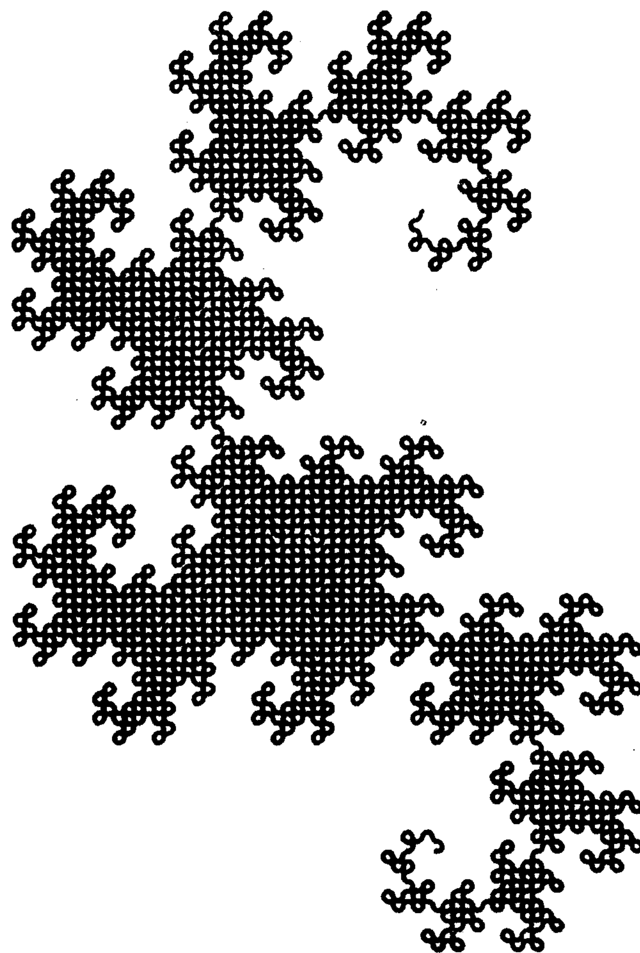


(1) NOUVEAU DRAGON

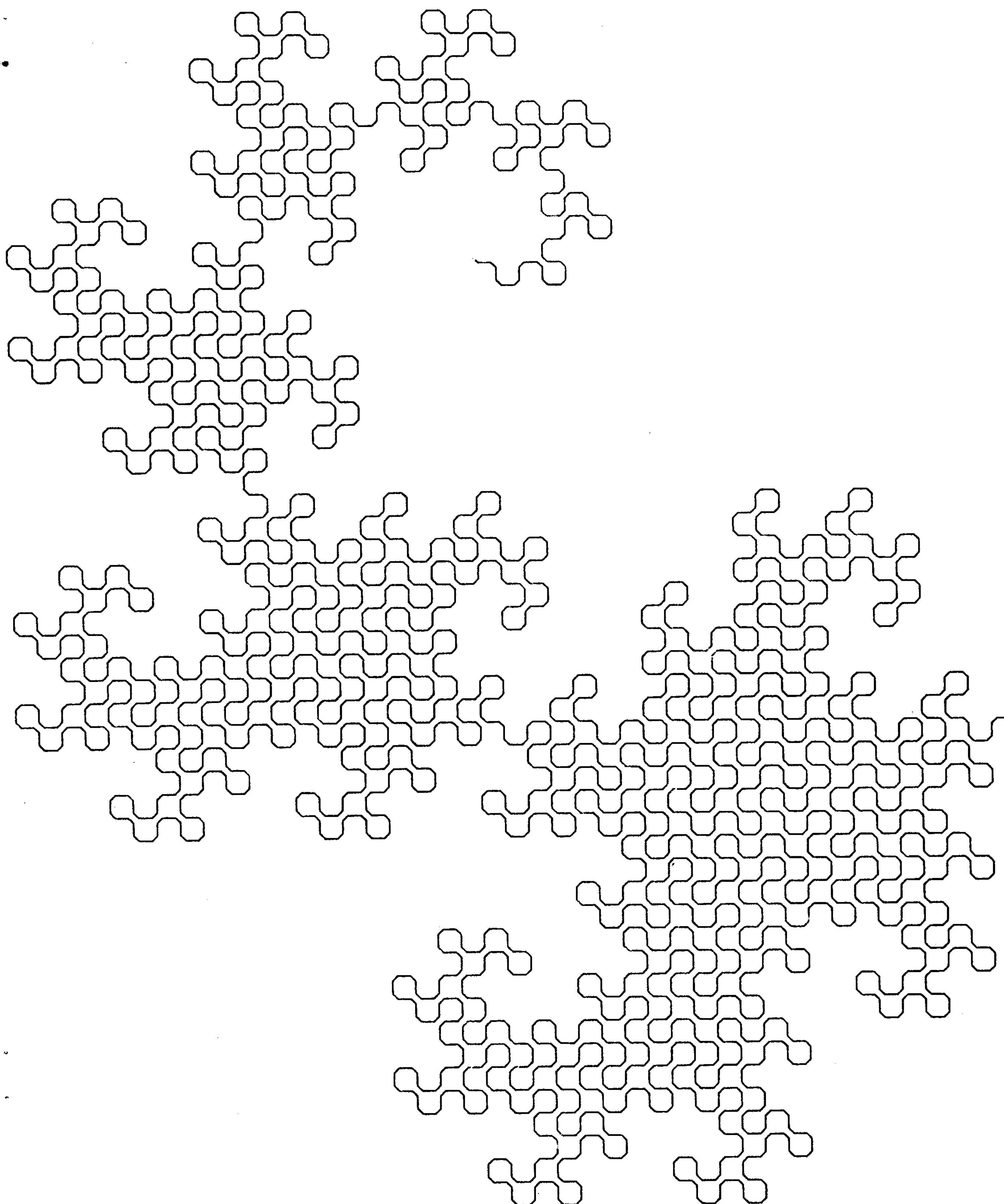
FFL: LAI 1
RSR

Début de la sortie binaire :

1	0	1	1	0	0	1	1	1	0	0	1	0	0	1	1	1	0	1	1	0	0	0	1	1
0	0	1	0	0	1	1	1	0	1	1	0	0	1	1	1	0	0	1	0	0	0	1	1	0
1	1	0	0	0	1	1	0	0	1	0	0	1	1	1	0	1	1	0	0	1	1	1	0	0
1	0	0	1	1	1	0	1	1	0	0	0	1	1	0	0	1	0	0	0	1	1	0	1	1
0	0	1	1	1	0	0	1	0	0	0	1	1	0	1	1	0	0	0	1	1	0	0	1	0
0	1	1	1	0	1	1	0	0	1	1	1	0	0	1	0	0	1	1	1	0	1	1	0	0
0	1	1	0	0	1	0	0	1	1	1	0	1	1	0	0	1	1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	0	1	1	0	0	1	0	0	0	1	1	0	1	1	0	0	1	1
1	0	0	1	0	0	1	1	1	0	1	1	0	0	0	1	1	0	0	1	0	0	0	1	1
0	1	1	0	0	1	1	1	0	0	1	0	0	0	1	1	0	1	1	0	0	0	1	1	0
0	1	0	0	1	1	1	0	1	1	0	0	1	1	1	0	0	1	0	0	1	1	1	0	1
1	0	0	0	1	1	0	0	1	0	0	1	1	1	0	1	1	0	0	1	1	1	0	0	1
0	0	0	1	1	0	1	1	0	0	0	1	1	0	0	1	0	0	1	1	1	0	1	1	0
0	1	1	1	0	0	1	0	0	1	1	1	0	1	1	0	0	0	1	1	0	0	1	0	0
0	1	1	0	1	1	0	0	1	1	1	0	0	1	0	0	0	1	1	0	1	1	0	0	0
1	1	0	0	1	0	0	0	1	1	0	1	1	0	0	1	1	1	0	0	1	0	0	1	1
1	0	1	1	0	0	0	1	1	0	0	1	0	0	1	1	1	0	1	1	0	0	1	1	1
0	0	1	0	0	0	1	1	0	1	1	0	0	0	1	1	0	0	1	0	0	0	1	1	0
1	1	0	0	1	1	1	0	0	1	0	0	1	1	1	0	1	1	0	0	0	1	1	0	0
1	0	0	0	1	1	0	1	1	0	0	1	1	1	0	0	1	0	0	0	1	1	0	1	1
0	0	0	1	1	0	0	1	0	0	1	1	1	0	1	1	0	0	1	1	1	0	0	1	0
0	1	1	1	0	1	1	0	0	0	1	1	0	0	1	0	0	1	1	1	0	1	1	0	0
1	1	1	0	0	1	0	0	0	1	1	0	1	1	0	0	0	1	1	0	0	1	0	0	1
1	1	0	1	1	0	0	1	1	1	0	0	1	0	0	1	1	1	0	1	1	0	0	0	1
1	0	0	1	0	0	0	1	1	0	1	1	0	0	1	1	1	0	0	1	0	0	0	1	1
0	1	1	0	0	0	1	1	0	0	1	0	0	1	1	1	0	1	1	0	0	1	1	1	0
0	1	0	0	1	1	1	0	1	1	0	0	0	1	1	0	0	1	0	0	1	1	1	0	1
1	0	0	1	1	1	0	0	1	0	0	0	1	1	0	1	1	0	0	0	1	1	0	0	1
0	0	0	1	1	0	1	1	0	0	1	1	1	0	0	1	0	0	1	1	1	0	1	1	0
0	0	1	1	0	0	1	0	0	0	1	1	0	1	1	0	0	1	1	1	0	0	1	0	0
0	1	1	0	1	1	0	0	0	1	1	0	0	1	0	0	0	1	1	0	1	1	0	0	1
1	1	0	0	1	0	0	1	1	1	0	1	1	0	0	0	1	1	0	0	1	0	0	1	1
1	0	1	1	0	0	1	1	1	0	0	1	0	0	0	1	1	0	1	1	0	0	0	1	1
0	0	1	0	0	1	1	1	0	1	1	0	0	1	1	1	0	0	1	0	0	1	1	1	0
1	1	0	0	0	1	1	0	0	1	0	0	0	1	1	0	1	1	0	0	1	1	1	0	0
1	0	0	0	1	1	0	1	1	0	0	0	1	1	0	0	1	0	0	0	1	1	0	1	1
0	0	1	1	1	0	0	1	0	0	1	1	1	0	1	1	0	0	0	1	1	0	0	1	0
0	1	1	1	0	1	1	0	0	1	1	1	0	0	1	0	0	0	1	1	0	1	1	0	0
0	1	1	0	0	1	0	0	0	1	1	0	1	1	0	0	1	1	1	0	0	1	0	0	1
1	1	0	1	1	0	0	0	1	1	0	0	1	0	0	0	1	1	0	1	1	0	0	1	1
1	0	0	1	0	0	0	1	1	0	1	1	0	0	0	1	1	0	0	1	0	0	1	1	1
0	1	1	0	0	1	1	1	0	0	1	0	0	1	1	1	0	1	1	0	0	0	1	1	0



Partie initiale agrandie

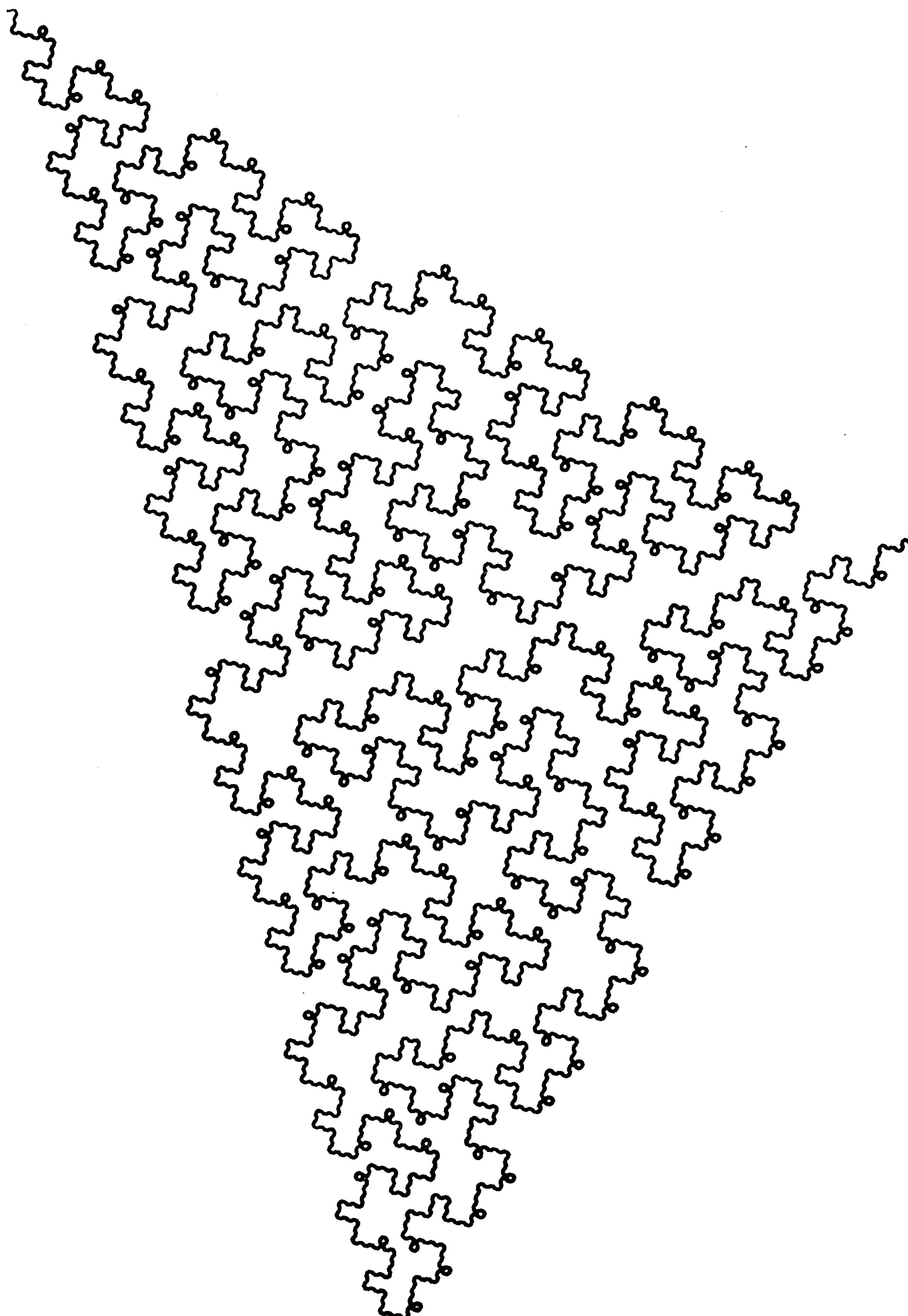


(2) FLIP-FLOP

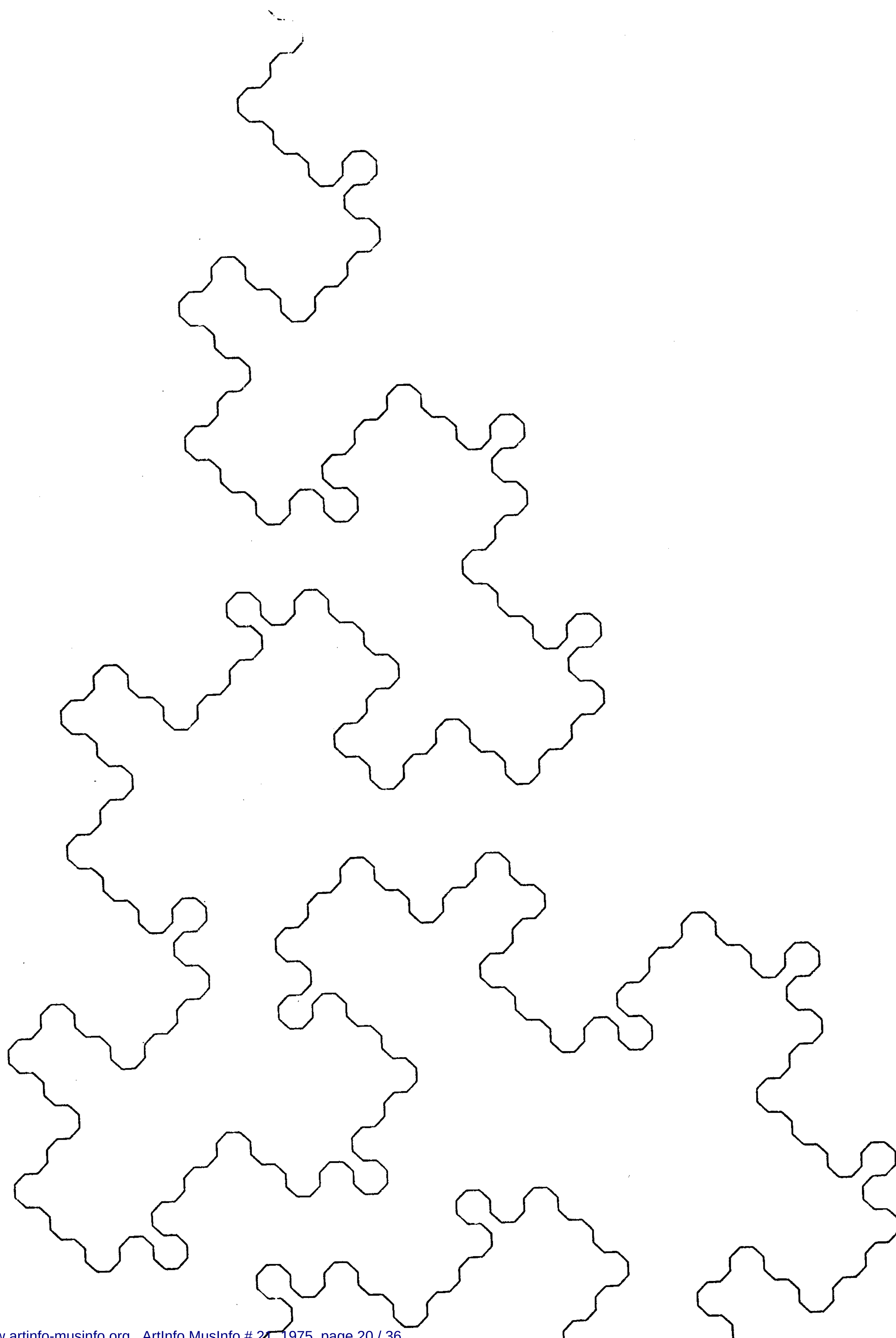
```
FFL:    LA  BITA
        IBT 15
        RSR
```

Début de la sortie binaire :

```
1 0 0 1 0 1 1 0 1 0 1 1 0 0 0 1 1 0 0 1 0 1 0 0 1
0 1 1 0 1 1 0 1 0 0 1 0 1 1 0 1 0 1 1 0 0 1 1 1 0
0 1 0 1 0 0 1 0 1 1 0 0 0 1 1 0 0 1 0 1 1 0 1 0 1
1 0 0 0 1 1 0 0 1 0 1 0 0 1 0 1 1 0 1 0 0 1 0 0 1
0 1 1 0 1 0 1 1 0 0 1 1 1 0 0 1 0 1 0 0 1 0 1 1 0
1 1 0 1 0 0 1 0 1 1 0 1 0 1 1 0 0 0 1 1 0 0 1 0 1
0 0 1 0 1 1 0 1 1 0 1 0 0 1 0 1 1 0 1 0 1 1 0 0 1
1 1 0 0 1 0 1 0 0 1 0 1 1 0 0 1 1 1 0 0 1 0 1 1 0
1 0 1 1 0 0 0 1 1 0 0 1 0 1 0 0 1 0 1 1 0 1 0 0 1
0 0 1 0 1 1 0 1 0 1 1 0 0 1 1 1 0 0 1 0 1 0 0 1 0
1 1 0 0 0 1 1 0 0 1 0 1 1 0 1 0 1 1 0 0 0 1 1 0 0
1 0 1 0 0 1 0 1 1 0 1 1 0 1 0 0 1 0 1 1 0 1 0 1 1
0 0 1 1 1 0 0 1 0 1 0 0 1 0 1 1 0 0 0 1 1 0 0 1 0
1 1 0 1 0 1 1 0 0 0 1 1 0 0 1 0 1 0 0 1 0 1 1 0 1
0 0 1 0 0 1 0 1 1 0 1 0 1 1 0 0 1 1 1 0 0 1 0 1 0
0 1 0 1 1 0 1 0 0 1 0 0 1 0 1 1 0 1 0 1 1 0 0 0 1
1 0 0 1 0 1 0 0 1 0 1 1 0 1 1 0 1 0 0 1 0 1 1 0 1
0 1 1 0 0 1 1 1 0 0 1 0 1 0 0 1 0 1 1 0 0 1 1 1 0
0 1 0 1 1 0 1 0 1 1 0 0 0 1 1 0 0 1 0 1 0 0 1 0 1
1 0 1 0 0 1 0 0 1 0 1 1 0 1 0 1 1 0 0 1 1 1 0 0 1
0 1 0 0 1 0 1 1 0 1 1 0 1 0 0 1 0 1 1 0 1 0 1 1 0
0 0 1 1 0 0 1 0 1 0 0 1 0 1 1 0 1 1 0 1 0 0 1 0 1
1 0 1 0 1 1 0 0 1 1 1 0 0 1 0 1 0 0 1 0 1 1 0 0 0
1 1 0 0 1 0 1 1 0 1 0 1 1 0 0 0 1 1 0 0 1 0 1 0 0
1 0 1 1 0 1 0 0 1 0 0 1 0 1 1 0 1 0 1 1 0 0 1 1 1
0 0 1 0 1 0 0 1 0 1 1 0 1 1 0 1 0 0 1 0 1 1 0 1 0
1 1 0 0 0 1 1 0 0 1 0 1 0 0 1 0 1 1 0 1 1 0 1 0 0
1 0 1 1 0 1 0 1 1 0 0 1 1 1 0 0 1 0 1 0 0 1 0 1 1
0 0 1 1 1 0 0 1 0 1 1 0 1 0 1 1 0 0 0 1 1 0 0 1 0
1 0 0 1 0 1 1 0 1 0 0 1 0 0 1 0 1 1 0 1 0 1 1 0 0
1 1 1 0 0 1 0 1 0 0 1 0 1 1 0 0 1 1 1 0 0 1 0 1 1
0 1 0 1 1 0 0 0 1 1 0 0 1 0 1 0 0 1 0 1 1 0 1 1 0
1 0 0 1 0 1 1 0 1 0 1 1 0 0 1 1 1 0 0 1 0 1 0 0 1
0 1 1 0 0 0 1 1 0 0 1 0 1 1 0 1 0 1 1 0 0 0 1 1 0
0 1 0 1 0 0 1 0 1 1 0 1 0 0 1 0 0 1 0 1 1 0 1 0 1
1 0 0 1 1 1 0 0 1 0 1 0 0 1 0 1 1 0 1 0 0 1 0 0 1
0 1 1 0 1 0 1 1 0 0 0 1 1 0 0 1 0 1 0 0 1 0 1 1 0
1 1 0 1 0 0 1 0 1 1 0 1 0 1 1 0 0 1 1 1 0 0 1 0 1
0 0 1 0 1 1 0 0 1 1 1 0 0 1 0 1 1 0 1 0 1 1 0 0 0
1 1 0 0 1 0 1 0 0 1 0 1 1 0 1 0 0 1 0 0 1 0 1 1 0
1 0 1 1 0 0 1 1 1 0 0 1 0 1 0 0 1 0 1 1 0 0 0 1 1
```



Partie initiale agrandie

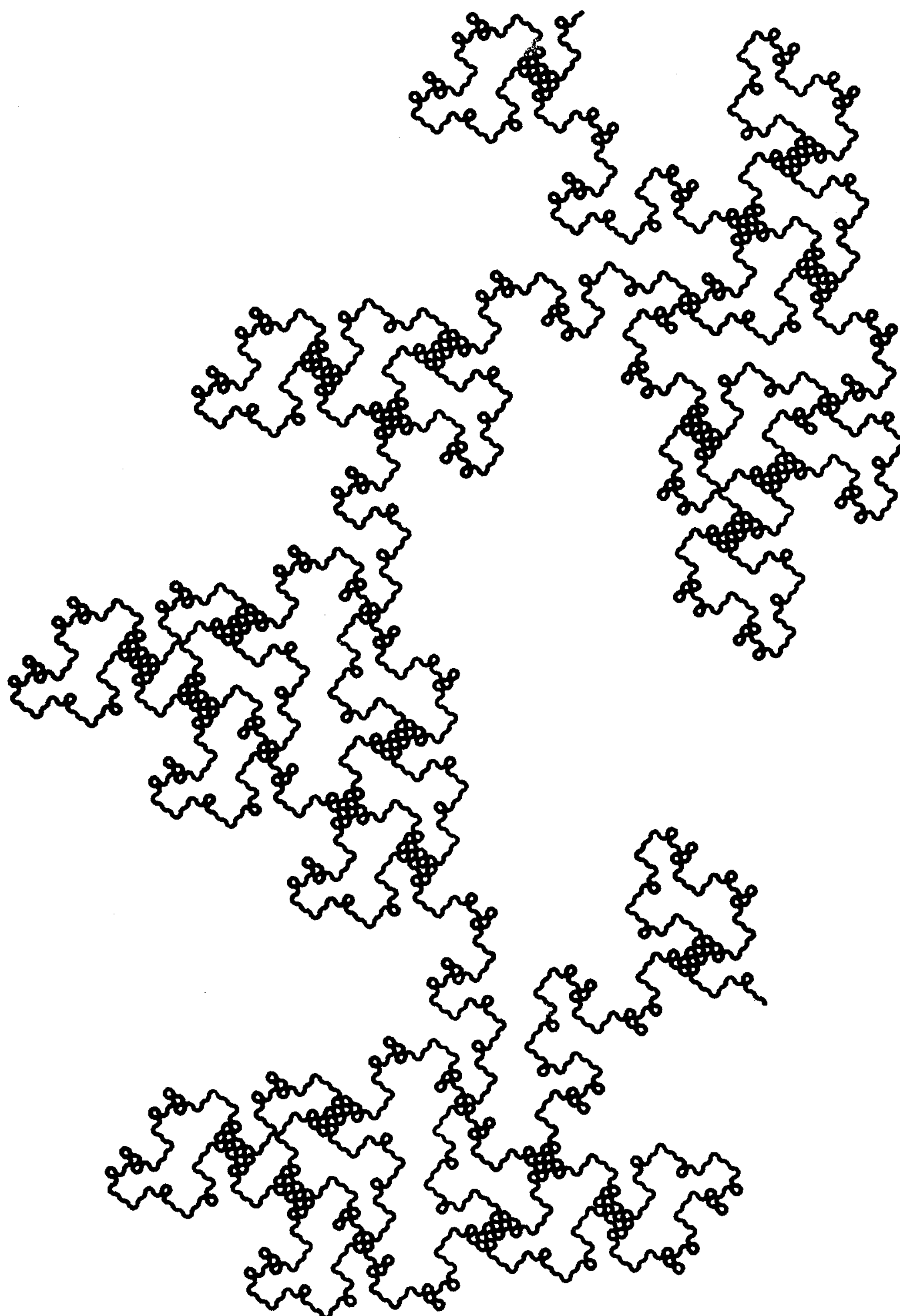


(3) SE BOUFFE LA QUEUE

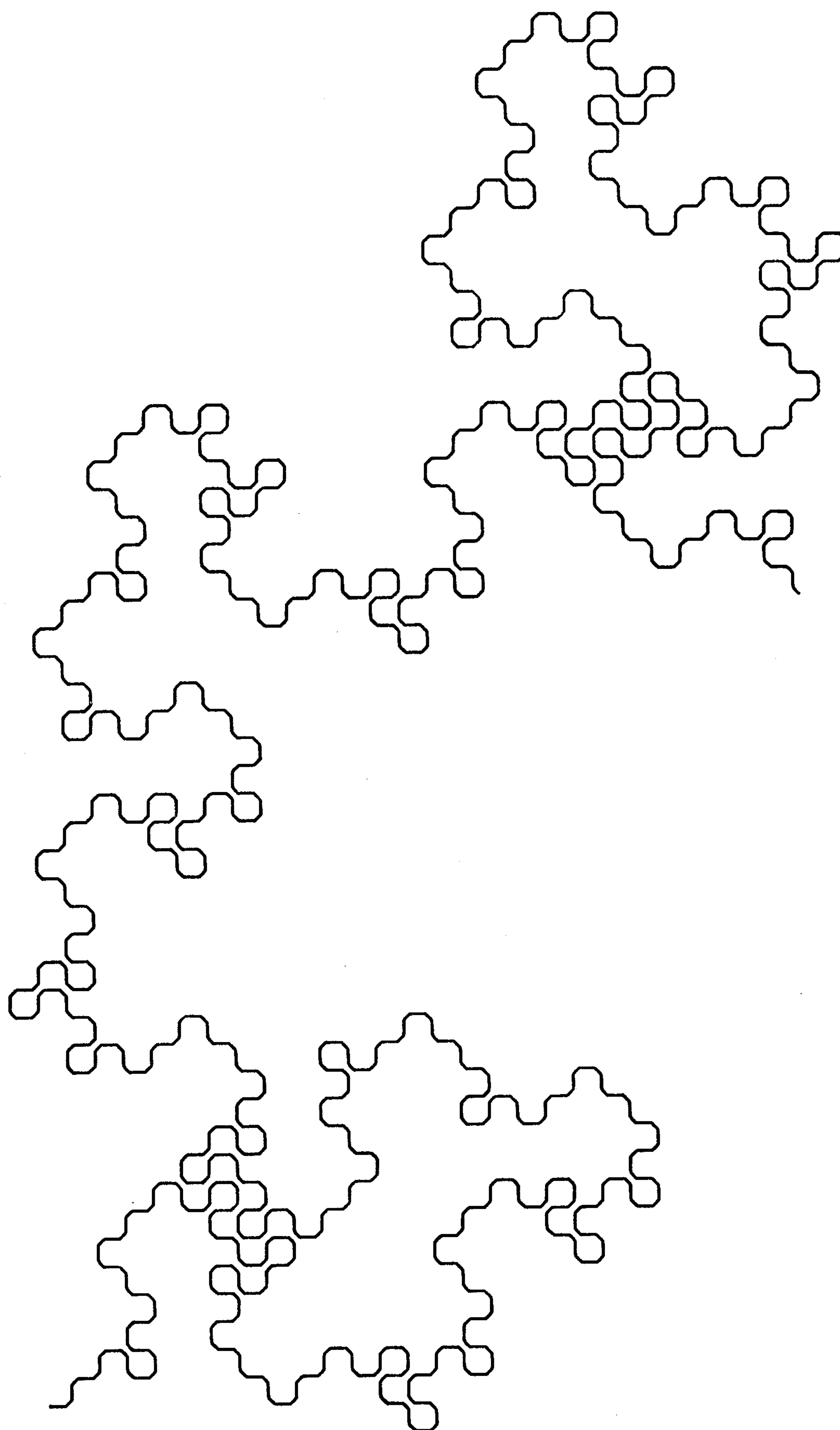
```
FFL:    LX    INDX
        LBY   &ATAB
        IC    INDX
        RSR
```

Début de la sortie binaire :

```
0 1 0 0 1 1 1 0 0 1 1 0 1 0 0 1 0 1 0 0 1 1 0 0 0
1 1 0 1 1 0 0 0 1 0 0 1 1 1 0 0 1 1 0 1 0 1 1 0 1
0 0 1 1 0 0 0 1 1 0 1 1 1 0 0 1 0 0 1 1 1 0 0 1 1
0 1 0 0 1 0 1 0 0 1 1 0 0 0 1 1 0 1 1 1 0 0 1 0 0
1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 0 1 1 0 0 0 1 1 0 1
0 1 1 0 1 0 0 1 1 1 0 0 1 1 0 1 0 0 1 0 1 0 0 1 1
0 0 0 1 1 0 1 1 0 0 0 1 0 0 1 1 1 0 0 1 1 0 1 0 1
1 0 1 0 0 1 1 0 0 0 1 1 0 1 1 0 0 0 1 0 0 1 1 1 0
0 1 1 0 1 0 0 1 0 1 0 0 1 1 0 0 0 1 1 0 1 1 1 0 0
1 0 0 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 0 1 1 0 0 0 1
1 0 1 0 1 1 0 1 0 0 1 1 1 0 0 1 1 0 1 0 0 1 0 1 0
0 1 1 0 0 0 1 1 0 1 1 0 0 0 1 0 0 1 1 1 0 0 1 1 0
1 0 1 1 0 1 0 0 1 1 0 0 0 1 1 0 1 1 1 0 0 1 0 0 1
1 1 0 0 1 1 0 1 0 0 1 0 1 0 0 1 1 0 0 0 1 1 0 1 1
1 0 0 1 0 0 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 0 1 1 0
0 0 1 1 0 1 0 0 1 0 1 0 0 1 1 1 0 0 1 1 0 1 0 0 1
0 1 0 0 1 1 0 0 0 1 1 0 1 1 0 0 0 1 0 0 1 1 1 0 0
1 1 0 1 0 1 1 0 1 0 0 1 1 0 0 0 1 1 0 1 1 0 0 0 1
0 0 1 1 1 0 0 1 1 0 1 0 0 1 0 1 0 0 1 1 0 0 0 1 1
0 1 1 1 0 0 1 0 0 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 0
1 1 0 0 0 1 1 0 1 0 0 1 0 1 0 0 1 1 1 0 0 1 1 0 1
0 0 1 0 1 0 0 1 1 0 0 0 1 1 0 1 1 0 0 0 1 0 0 1 1
1 0 0 1 1 0 1 0 1 1 0 1 0 0 1 1 0 0 0 1 1 0 1 1 1
0 0 1 0 0 1 1 1 0 0 1 1 0 1 0 0 1 0 1 0 0 1 1 0 0
0 1 1 0 1 1 1 0 0 1 0 0 1 1 1 0 0 1 1 0 1 0 1 1 0
1 0 0 1 1 0 0 0 1 1 0 1 0 1 1 0 1 0 0 1 1 1 0 0 1
1 0 1 0 0 1 0 1 0 0 1 1 0 0 0 1 1 0 1 1 0 0 0 1 0
0 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 0 1 1 0 0 0 1 1 0
1 1 0 0 0 1 0 0 1 1 1 0 0 1 1 0 1 0 0 1 0 1 0 0 1
1 0 0 0 1 1 0 1 1 1 0 0 1 0 0 1 1 1 0 0 1 1 0 1 0
1 1 0 1 0 0 1 1 0 0 0 1 1 0 1 0 0 1 0 1 0 0 1 1 1
0 0 1 1 0 1 0 0 1 0 1 0 0 1 1 0 0 0 1 1 0 1 1 0 0
0 1 0 0 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 0 1 1 0 0 0
1 1 0 1 1 1 0 0 1 0 0 1 1 1 0 0 1 1 0 1 0 0 1 0 1
0 0 1 1 0 0 0 1 1 0 1 1 1 0 0 1 0 0 1 1 1 0 0 1 1
0 1 0 1 1 0 1 0 0 1 1 0 0 0 1 1 0 1 0 0 1 0 1 0 0
1 1 1 0 0 1 1 0 1 0 0 1 0 1 0 0 1 1 0 0 0 1 1 0 1
1 0 0 0 1 0 0 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 0 1 1
0 0 0 1 1 0 1 1 0 0 0 1 0 0 1 1 1 0 0 1 1 0 1 0 0
1 0 1 0 0 1 1 0 0 0 1 1 0 1 1 1 0 0 1 0 0 1 1 1 0
```



Partie initiale agrandie



(4) SE BOUFFE LA QUEUE DE 2 EN 2

```
FFL:    LX    INDX
        LBY   &ATAB
        LYI   2
        ADR   Y,X
        STX   INDX
        RSR
```

Début de la sortie binaire :

```
0 1 1 0 1 0 0 1 0 1 0 0 1 1 0 0 0 1 1 0 1 0 1 1 0
1 0 0 1 1 1 0 0 1 1 0 1 0 0 1 0 1 0 0 1 1 1 0 0 1
1 0 1 0 1 1 0 1 0 0 1 0 1 1 0 1 1 0 1 0 0 1 0 1 0
0 1 1 0 0 0 1 1 0 1 0 1 1 0 1 0 0 1 1 0 0 0 1 1 0
1 0 0 1 0 1 0 0 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 0 1
0 0 1 0 1 1 0 1 0 0 1 0 1 0 0 1 1 0 0 0 1 1 0 1 0
1 1 0 1 0 0 1 1 1 0 0 1 1 0 1 0 0 1 0 1 0 0 1 1 1
0 0 1 1 0 1 0 1 1 0 1 0 0 1 0 0 1 0 1 1 0 1 0 0 1
0 1 0 0 1 1 0 0 0 1 1 0 1 0 1 1 0 1 0 0 1 1 0 0 0
1 1 0 1 0 0 1 0 1 0 0 1 1 1 0 0 1 1 0 1 0 1 1 0 1
0 0 1 1 1 0 0 1 1 0 1 0 0 1 0 1 0 0 1 1 0 0 0 1 1
0 1 0 1 1 0 1 0 0 1 1 1 0 0 1 1 0 1 0 0 1 0 1 0 0
1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 0 1 0 1 1 0 1 1 0 1
0 0 1 0 1 0 0 1 1 0 0 0 1 1 0 1 0 1 1 0 1 0 0 1 1
0 0 0 1 1 0 1 0 0 1 0 1 0 0 1 1 1 0 0 1 1 0 1 0 1
1 0 1 0 0 1 0 1 1 0 1 1 0 1 0 0 1 0 1 0 0 1 1 0 0
0 1 1 0 1 0 1 1 0 1 0 0 1 1 1 0 0 1 1 0 1 0 0 1 0
1 0 0 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 0 1 0 0 1 0 1
1 0 1 0 0 1 0 1 0 0 1 1 0 0 0 1 1 0 1 0 1 1 0 1 0
0 1 1 0 0 0 1 1 0 1 0 0 1 0 1 0 0 1 1 1 0 0 1 1 0
1 0 1 1 0 1 0 0 1 0 0 1 0 1 1 0 1 0 0 1 0 1 0 0 1
1 0 0 0 1 1 0 1 0 1 1 0 1 0 0 1 1 1 0 0 1 1 0 1 0
0 1 0 1 0 0 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 0 1 0 1
1 0 1 1 0 1 0 0 1 0 1 0 0 1 1 0 0 0 1 1 0 1 0 1 1
0 1 0 0 1 1 0 0 0 1 1 0 1 0 0 1 0 1 0 0 1 1 1 0 0
1 1 0 1 0 1 1 0 1 0 0 1 0 0 1 0 1 1 0 1 0 0 1 0 1
0 0 1 1 0 0 0 1 1 0 1 0 1 1 0 1 0 0 1 1 1 0 0 1 1
0 1 0 0 1 0 1 0 0 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 0
1 0 0 1 0 1 1 0 1 0 0 1 0 1 0 0 1 1 0 0 0 1 1 0 1
0 1 1 0 1 0 0 1 1 0 0 0 1 1 0 1 0 0 1 0 1 0 0 1 1
1 0 0 1 1 0 1 0 1 1 0 1 0 0 1 1 0 0 0 1 1 0 1 0 0
1 0 1 0 0 1 1 0 0 0 1 1 0 1 0 1 1 0 1 0 0 1 1 1 0
0 1 1 0 1 0 0 1 0 1 0 0 1 1 1 0 0 1 1 0 1 0 1 1 0
1 0 0 1 0 1 1 0 1 1 0 1 0 0 1 0 1 0 0 1 1 0 0 0 1
1 0 1 0 1 1 0 1 0 0 1 1 0 0 0 1 1 0 1 0 0 1 0 1 0
0 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 0 1 0 1 1 0 1 1 0
1 0 0 1 0 1 0 0 1 1 0 0 0 1 1 0 1 0 1 1 0 1 0 0 1
1 1 0 0 1 1 0 1 0 0 1 0 1 0 0 1 1 1 0 0 1 1 0 1 0
1 1 0 1 0 0 1 0 0 1 0 1 1 0 1 0 0 1 0 1 0 0 1 1 0
0 0 1 1 0 1 0 1 1 0 1 0 0 1 1 0 0 0 1 1 0 1 0 0 1
0 1 0 0 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 0 1 1 1 0 0
1 1 0 1 0 0 1 0 1 0 0 1 1 0 0 0 1 1 0 1 0 1 1 0 1
0 0 1 1 1 0 0 1 1 0 1 0 0 1 0 1 0 0 1 1 1 0 0 1 1
0 1 0 1 1 0 1 0 0 1 0 1 1 0 1 1 0 1 0 0 1 0 1 0 0
1 1 0 0 0 1 1 0 1 0 1 1 0 1 0 0 1 1 0 0 0 1 1 0 1
0 0 1 0 1 0 0 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 0 1 0
0 1 0 1 1 0 1 0 0 1 0 1 0 0 1 1 0 0 0 1 1 0 1 0 1
1 0 1 0 0 1 1 1 0 0 1 1 0 1 0 0 1 0 1 0 0 1 1 1 0
0 1 1 0 1 0 1 1 0 1 0 0 1 0 0 1 0 1 1 0 1 0 0 1 0
1 0 0 1 1 0 0 0 1 1 0 1 0 1 1 0 1 0 0 1 1 0 0 0 1
1 0 1 0 0 1 0 1 0 0 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0
0 1 1 1 0 0 1 1 0 1 0 0 1 0 1 0 0 1 1 0 0 0 1 1 0
1 0 1 1 0 1 0 0 1 1 1 0 0 1 1 0 1 0 0 1 0 1 0 0 1
1 1 0 0 1 1 0 1 0 1 1 0 1 0 0 1 0 1 1 0 1 1 0 1 0
```

